|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**  **TP. HỒ CHÍ MINH** |

**BÁO CÁO MÔN HỌC**

**Tên học phần: Các hệ thống máy tính điều khiển**

*Kỳ thi học kỳ 2 đợt A năm học 2023 -2024*

**Giảng viên hướng dẫn: Phạm Quốc Phương**

**Tên đề tài:**

**Nhận diện biển báo**

**giao thông bằng**

**yolov8**

**Sinh viên thực hiện:**

**Nhóm 9**

**Nguyễn Văn Đạt- Mã số sinh viên: 2286300010**

**Ngành: Robot và Trí tuệ nhân tạo**

**Khoa/Viện: Kỹ thuật**

*Tp.HCM, ngày tháng năm 2024­­*

CÂU 1. ( 5.0 điểm ) (CLO: CLO3)

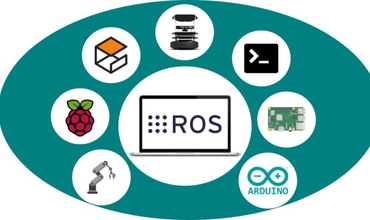
a. Hệ điều hành ROS là gì? Các bước cài đặt HĐH ROS. 2.0đ

b. Liệt kê một số giao thức truyền thông trong các hệ thống máy tính điều khiển ? 1.0đ

c. Phân biệt MPU và MCU? 1.0đ

d. Trình bày các ứng dụng Robot tích hợp AI trong thực tế 1.0đ

**a. Hệ điều hành ROS là gì?**

Robot Operation System (ROS) là một framework để viết ứng dụng robot. Nó là một tập hợp các công cụ, thư viện và quy ước nhằm mục đích đơn giản hóa nhiệm vụ tạo ra các hành vi phức tạp và mạnh mẽ của robot trên nhiều nền tảng robot khác nhau.

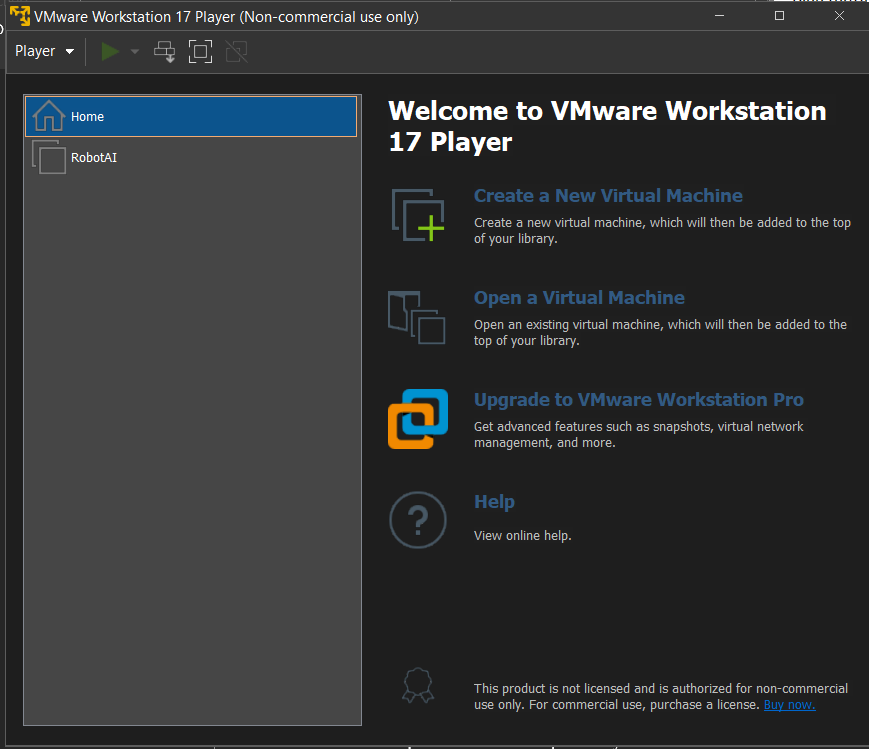
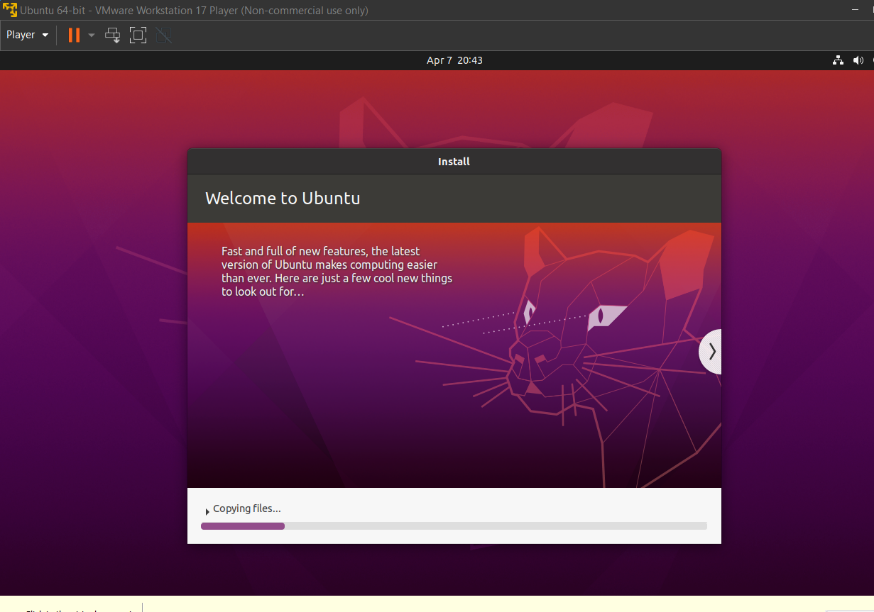
Các bước cái đặt HĐH ROS:

Dưới đây là các bước cài đặt ROS trên Ubuntu sử dụng máy ảo VMWare:

1. Chuẩn bị máy ảo VMWare:

Tải và cài đặt VMWare trên máy tính của bạn.

Tạo máy ảo mới ( Ở đây đặt tên là RobotAI) và cài đặt Ubuntu trên đó.



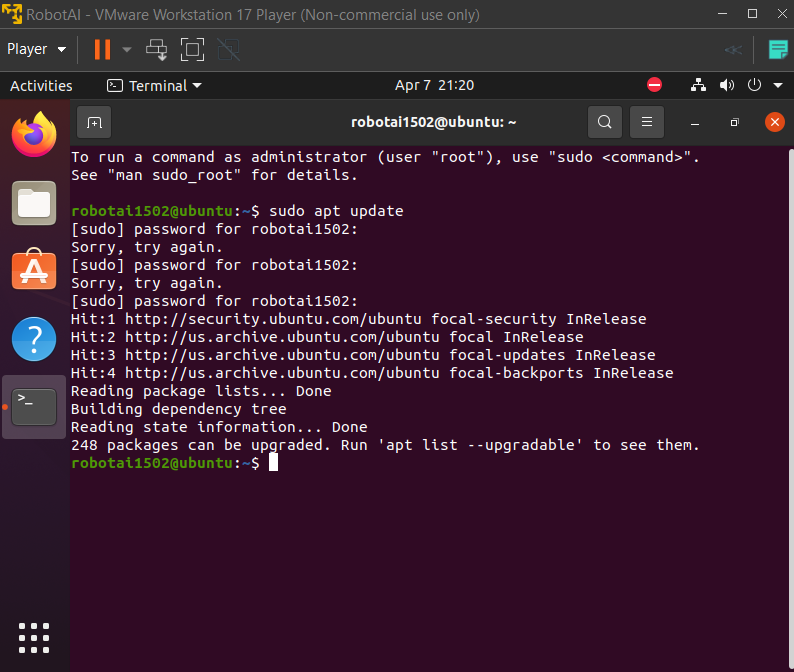
2. Cập nhật hệ thống Ubuntu:

- Mở terminal trên Ubuntu.

- Chạy các lệnh sau để cập nhật hệ thống:

sudo apt update

sudo apt upgrade



3. Cài đặt ROS:

- Mở terminal và thêm kho lưu trữ của ROS vào danh sách các nguồn mà Ubuntu cập nhật từ đó.

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

- Thêm khóa của kho lưu trữ:

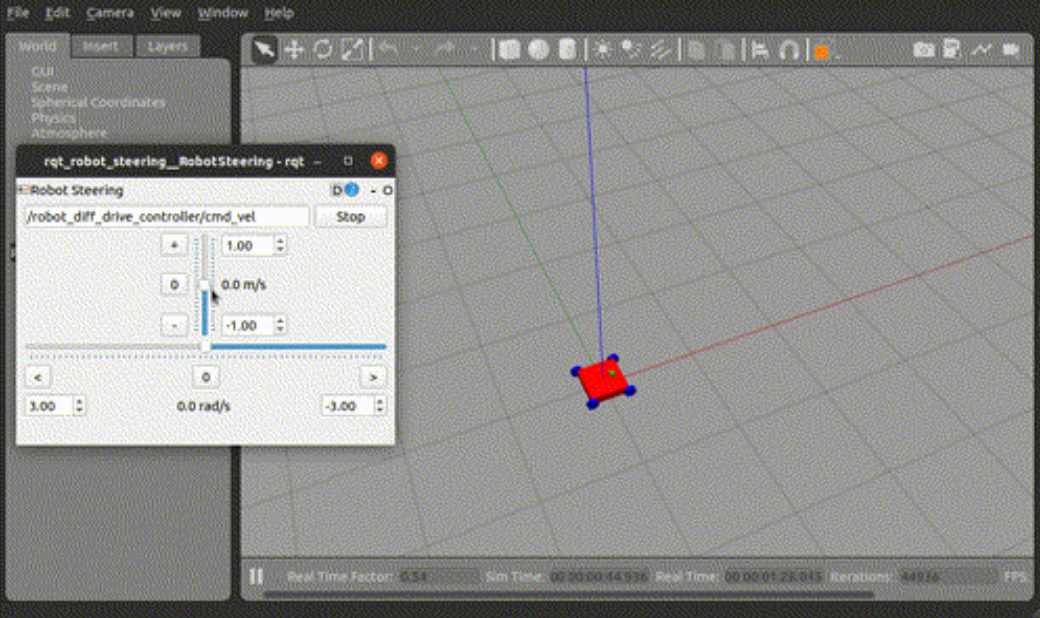
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add –

- Cập nhật lại danh sách gói và cài đặt ROS:

sudo apt update

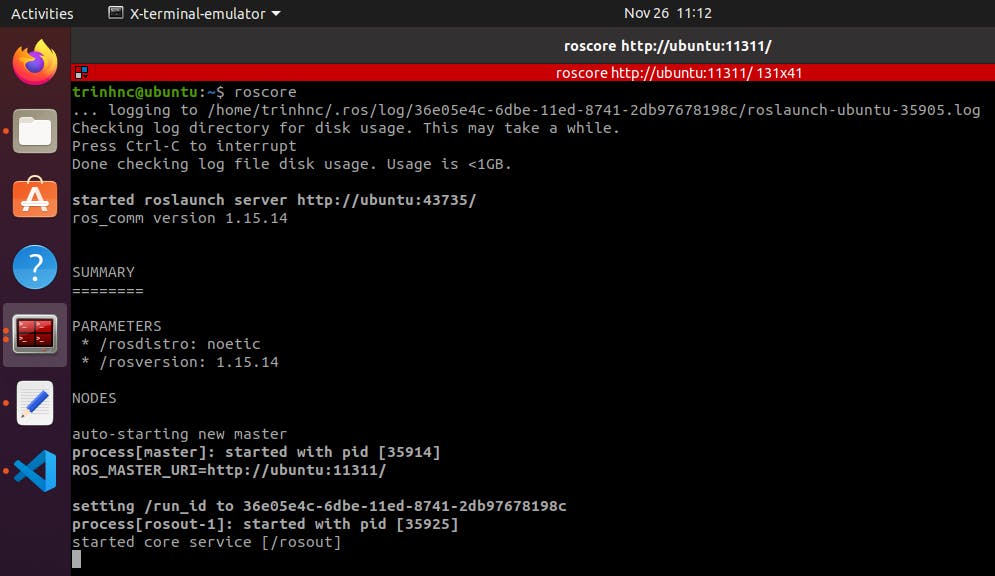
sudo apt install ros-noetic-desktop-full

4. Khởi động ROS:



- Chạy lệnh sau để khởi động ROS:

source /opt/ros/melodic/setup.bash



5. Cài đặt các gói phụ trợ và công cụ phát triển:

- Cài đặt các gói phụ trợ cho ROS:

sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool build-essential

- Khởi tạo rosdep:

sudo apt install python-rosdep

sudo rosdep init

rosdep update

6. Cấu hình môi trường làm việc:

- Chạy lệnh sau để tự động thêm dòng lệnh khởi động ROS vào tệp `.bashrc`:

echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc

source ~/.bashrc

7. Kiểm tra cài đặt:

- Chạy lệnh sau để kiểm tra việc cài đặt ROS đã thành công:

roscore

- Nếu mọi thứ chạy đúng, bạn sẽ thấy các thông báo hiển thị trong terminal.

**b. Liệt kê một số giao thức truyền thông trong các hệ thống máy tính điều khiển**

Dưới đây là một số giao thức truyền thông phổ biến trong các hệ thống máy tính điều khiển:

|  |  |
| --- | --- |
| 1. TCP/IP (Transmission Control Protocol/Internet Protocol): | Đây là giao thức truyền thông mạng phổ biến nhất trong hệ thống máy tính điều khiển. TCP/IP chịu trách nhiệm cho việc phân gói, gửi và nhận dữ liệu qua mạng Internet. Nó cung cấp một cơ chế tin cậy để truyền dữ liệu giữa các thiết bị trên mạng. |
| 2. UART (Universal Asynchronous Receiver/Transmitter): | Giao thức này được sử dụng để truyền dữ liệu giữa các thiết bị điện tử thông qua các đường dây hoặc kết nối không dây như Bluetooth hoặc IrDA. UART hoạt động theo cơ chế không đồng bộ, tức là không có tín hiệu đồng hồ chung giữa các thiết bị truyền và nhận. |
| 3. I2C (Inter-Integrated Circuit): | Giao thức I2C được sử dụng để kết nối các linh kiện điện tử trong một hệ thống, như các cảm biến và vi điều khiển. I2C cho phép nhiều thiết bị chia sẻ cùng một đường truyền dữ liệu và sử dụng hai dây (đồng hồ và dữ liệu) để truyền thông. |
| 4. SPI (Serial Peripheral Interface): | Giao thức SPI cung cấp một cách để truyền dữ liệu giữa các linh kiện điện tử trong một hệ thống. Nó sử dụng một loạt các tín hiệu đồng hồ và dữ liệu song song để truyền dữ liệu với tốc độ nhanh và không đồng bộ. |
| 5. CAN (Controller Area Network): | CAN được sử dụng chủ yếu trong các ứng dụng điều khiển ô tô và các hệ thống máy tính điều khiển khác. Giao thức này được thiết kế để chịu được nhiễu và cung cấp khả năng truyền thông đồng thời giữa nhiều thiết bị. |
| 6. Modbus: | Giao thức Modbus thường được sử dụng trong các hệ thống điều khiển công nghiệp để truyền dữ liệu giữa các thiết bị như cảm biến và bộ điều khiển. Modbus có thể hoạt động trên nhiều phương tiện truyền thông, bao gồm RS-232, RS-485 và TCP/IP. |

Các giao thức trên đều có vai trò quan trọng trong việc truyền thông và tương tác giữa các thành phần trong các hệ thống máy tính điều khiển.

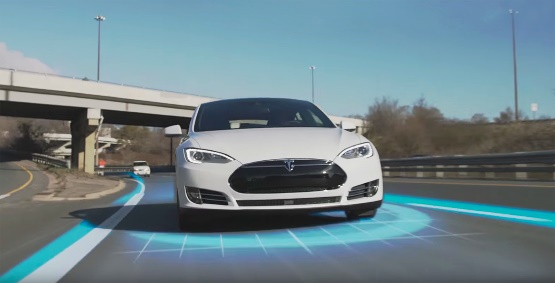
**c. Phân biệt MPU và MCU:**

MPU (Microprocessor Unit) và MCU (Microcontroller Unit) đều là loại vi xử lý nhúng nhưng có sự khác biệt về cấu trúc và ứng dụng chính:

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **MPU** | **MCU** |
| Tên đầy đủ | Microprocessor Unit | Microcontroller Unit |
| **Thiết kế** | Thường đi kèm với bộ nhớ RAM, ROM, I/O, và các thanh ghi | Tích hợp CPU, bộ nhớ, I/O và các tính năng khác trên một chip duy nhất |
| **Chức năng chính** | Xử lý dữ liệu và thực hiện các tác vụ phức tạp.  MPU là một vi xử lý tổng hợp dành cho các ứng dụng đa nhiệm, có khả năng xử lý tác vụ phức tạp và thường chạy hệ điều hành như Linux hoặc Windows. | Điều khiển các thiết bị nhúng và thực hiện các tác vụ điều khiển.  MCU là một vi xử lý nhỏ gọn tích hợp trên một chip, thường không chạy hệ điều hành và được sử dụng trong các ứng dụng nhúng đơn giản. |
| **Công nghệ sử dụng** | Thường sử dụng kiến trúc RISC hoặc CISC | Thường sử dụng kiến trúc RISC hoặc CISC |
| **Thích hợp cho ứng dụng** | Thích hợp cho các ứng dụng yêu cầu xử lý cao và tính linh hoạt cao | Thích hợp cho các ứng dụng nhúng đơn giản đến trung bình |
| **Tiêu thụ điện năng** | Tiêu thụ năng lượng cao | Tiêu thụ năng lượng thấp hơn |
| **Chi phí** | Thường có giá thành cao hơn | Thường có giá thành thấp hơn |
| **Kích thước** | Thường có kích thước lớn hơn và phức tạp | Thường có kích thước nhỏ hơn và ít phức tạp |

**d. Trình bày các ứng dụng Robot tích hợp AI trong thực tế:**

Các ứng dụng của robot tích hợp trí tuệ nhân tạo (AI) trong thực tế rất đa dạng và ngày càng được triển khai rộng rãi. Dưới đây là một số ứng dụng robot tích hợp AI trong thực tế:



1. Xe tự lái: Các công ty như Tesla, Waymo và Uber đã phát triển các công nghệ xe tự lái sử dụng AI để tự động lái xe và quản lý giao thông. Hệ thống AI của các xe tự lái có khả năng nhận diện và phản ứng với các tình huống phức tạp trên đường, như đèn giao thông, biển báo và các phương tiện khác.



2. Robot dịch vụ: Các robot dịch vụ như Pepper của Softbank hoặc Spot của Boston Dynamics được thiết kế để hỗ trợ trong các môi trường như khách sạn, nhà hàng, bệnh viện và văn phòng. Các robot này có khả năng tương tác với con người, nhận diện giọng nói và khuôn mặt, cũng như thực hiện các nhiệm vụ như hướng dẫn, giao tiếp và giải trí.



3. Robot y tế: Trong lĩnh vực y tế, robot tích hợp AI được sử dụng để hỗ trợ trong các tác vụ như phẫu thuật, chẩn đoán bệnh và chăm sóc bệnh nhân. Các robot như da Vinci Surgical System được sử dụng để thực hiện các ca phẫu thuật với độ chính xác cao hơn và ít biến chứng hơn so với phẫu thuật thủ công.



4. Robot sản xuất: Trong ngành công nghiệp, robot được tích hợp AI được sử dụng để tự động hóa các quy trình sản xuất và lắp ráp. Các robot có khả năng học và điều chỉnh theo môi trường làm việc, tối ưu hóa quy trình sản xuất và giảm thời gian chết.



5. Robot giáo dục: Trong giáo dục, robot được sử dụng để hỗ trợ trong việc giảng dạy và học tập. Các robot giáo dục như Cozmo hoặc Nao được sử dụng để giảng dạy các kỹ năng STEM cho trẻ em, khuyến khích sự tò mò và tương tác thông qua các hoạt động lập trình và giải đố.



6. Robot an ninh: Trong lĩnh vực an ninh và giám sát, robot được sử dụng để patrolling, giám sát và phát hiện các hành vi đe dọa. Các robot an ninh có thể sử dụng AI để nhận diện khuôn mặt, phát hiện vật thể nghi ngờ và gửi cảnh báo đến các trạm kiểm soát.

Những ứng dụng trên chỉ là một phần nhỏ của cách mà robot tích hợp AI được triển khai trong thực tế. Trong tương lai, dự kiến ​​sự phát triển của công nghệ AI sẽ mở ra nhiều cơ hội mới và mở rộng các ứng dụng của robot trong nhiều lĩnh vực khác nhau.

CÂU 2. (5.0 điểm ) (CLO: CLO4)

Sinh viên tự chọn 1 project: “Ứng dụng máy tính điều khiển trong thực tế”

Yêu cầu:

- Hardware: Raspberry pi, Jetson nano

- Phần mềm: Linux, Raspbian, ROS,…

- Ngoại vi: Cảm biến, led, nút nhấn, RC Servo, DC Motor, Camera,…

**Project: “Nhận diện biển báo giáo thông sử dụng Yolov8 và điểu khiển led qua Arduino”**

**I.Yêu cầu:**

- **Hardware**: Windows

- **Phần mềm cần thiết:**

Python: Để viết mã cho việc nhận diện biển báo và hệ thống đèn giao thông.

YOLOv8: Một mô hình nhận diện vật thể (object detection) sử dụng deep learning.

Arduino IDE: Để lập trình và điều khiển Arduino.

Ổ đĩa Arduino: Driver cho board Arduino để giao tiếp với máy tính qua cổng USB.

Visual Studio Code: Để viết và chạy mã Python.

**- Ngoại vi cần thiết:**

Arduino Uno hoặc tương tự: Board Arduino để điều khiển các đèn LED.

LEDs và các phụ kiện kết nối (resistor, dây dẫn, breadboard, etc.): Để tạo hệ thống đèn giúp phát hiện biển báo.

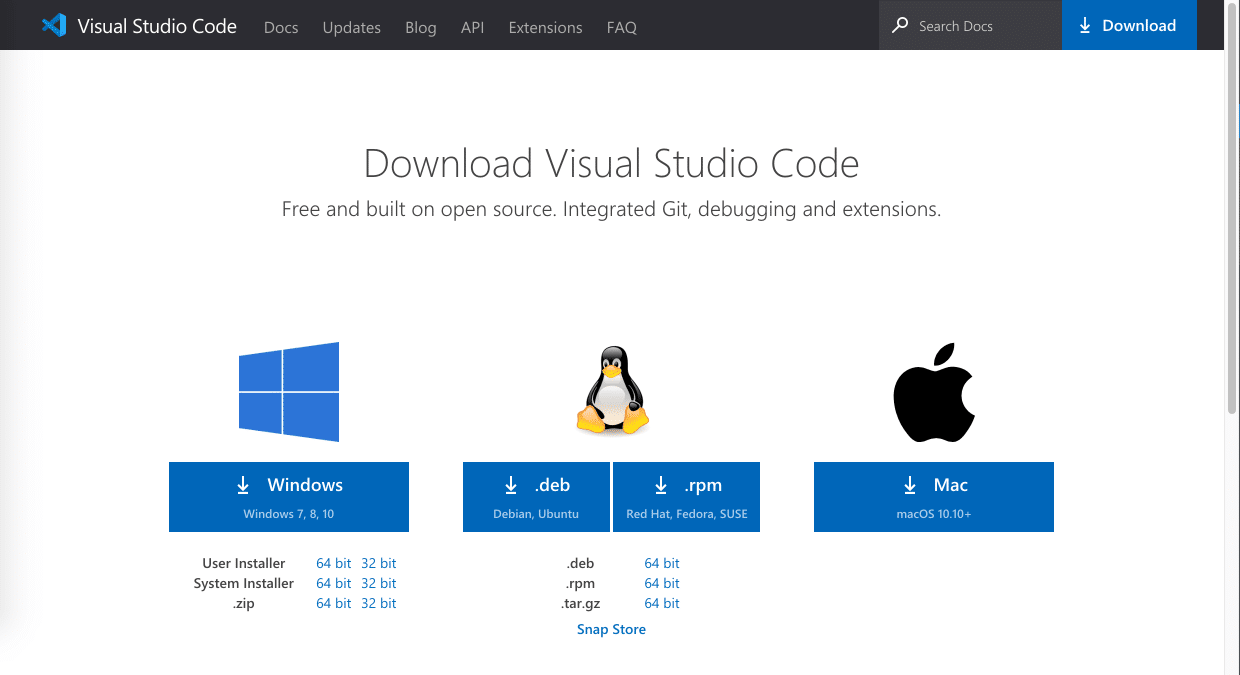
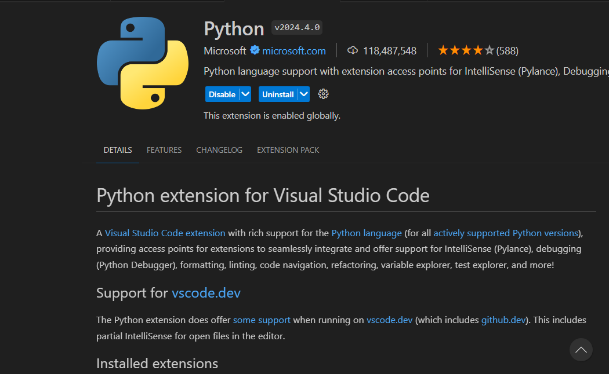
Camera hoặc webcam: Để cung cấp hình ảnh cho việc nhận diện biển báo và hệ thống biển báo giao thông.

Cáp USB: Để kết nối Arduino với máy tính.

Tóm Tắt: Trong dự án này python sẽ là ngôn ngữ lập trình chính dùng để mở camare hay hình ảnh để nhận diện từ đó có thể truyền tín hiệu đến Arduino để điều khiển led đưa ra các tín hiệu khác nhau để phân biệt từng hình ảnh được đưa ra. Dự án này sẽ nhận diện các biển báo giao thông và từ đó điều khiển led như là tín hiệu nhận về để cho người dùng biết được là biển báo đó có cho xe tiếp tục đi hay không hoặc đi chậm trên một đoạn đường nào đó. Ví dụ: khi một chiếc xe tự hành với khối lượng là 5 tấn đi vào đường một chiều và nhận diện được biển báo cấm đi ngược chiều thì xe sẽ bật đèn đỏ để người dùng biết được rằng xe sẽ không thể tiếp tục đi được, khi đi đến đoạn đường cảnh báo có người đi qua đường thì xe sẽ bật đèn vàng để người dùng biết được xe sẽ đi chậm trên đoạn đường đó, và khi xe đi qua đoạn đường có biển báo giới hạn 10 tấn nhưng xe chỉ 5 tấn thì xe sẽ bật đèn xanh dể người dùng biết được xe sẽ tiếp tục đi mà không gặp trở ngại.

**II. Phương Pháp:**

Đầu tiên cần tải VS code vào hệ điều hành chính Windows, đây là một phần mềm đa chức năng và chuyên nghiệp trong lập trình bao gồm nhiều ngôn ngữ khác nhau, và đây cũng là phần mềm chính để lập trình python trong dự án này:



Giao diện cài đặt Vscode và Python trong Vscode

**Các thư viện chính trong Python:**

* OpenCV: Thư viện xử lý ảnh và video
* Arduino-Python API: Thư viện Python để giao tiếp với Arduino thông qua serial port.
* Ultralytics: là thư viện được sử dụng để tải mô hình trọng số của YOLOv8
* Serial: là thư viện để có thể giao tiếp với arduino để điều khiển thiết vị ngoại vi (LED)

Các bước thực hiện:

**Bước 1: Chuẩn bị môi trường phần mềm**

* Cài đặt Python trên máy tính.
* Cài đặt OpenCV và các thư viện phụ trợ thông qua pip.
* Tải xuống và cài đặt Arduino IDE từ trang web chính thức của Arduino.
* Cài đặt thư viện Arduino-Python API.
* Tải và cài đặt Ultralytics-YOLOv8.

**Bước 2: Kết nối Arduino và LEDs**

Kết nối các LED với arduino đúng với số chân digital. Ở đây các các đén được mắc với sơ đồ mạch lần lượt là: đèn đỏ ( chân số 2 ), đen vàng ( chân số 3), đèn xanh ( chân số 4), với mỗi led sẽ có một điện trở giúp bảo vệ led khỏi quá tải và hoạt động ổn định hơn.

Kết nối Arduino với máy tính thông qua cáp USB.

**Bước 3: Sử dụng thuật toán YOLOV8 để huấn luyện tệp dataset đã chuẩn bị sẵn**

Dự án này sẽ sử dụng kiến trúc mạng nơ-ron YOLOv8 (You Only Look Once version 8) để nhận diện các biển báo giao thông trên hình ảnh. YOLO là một mô hình nhận diện đối tượng thời gian thực được thiết kế để xác định và phân loại các đối tượng trong một hình ảnh một cách hiệu quả và nhanh chóng.

Dữ liệu để thực hiện nhiệm vụ nhận diện biển báo giao thông cho dự án được tải từ nhiều nguồn khác nhau với định dạng YOLOv8 để tiến hành huấn luyện dữ liệu. Việc huấn luyện mô hình YOLOv8 trên một tập dữ liệu lớn chứa hình ảnh các biển báo giao thông, được gán nhãn đúng cho mỗi loại biển báo, sẽ tạo ra một mô hình có khả năng nhận diện các biển báo giao thông với độ chính xác cao.

Sau khi đoạn tệp đã huấn luyện xong giúp nhận diện hình ảnh tốt hơn, thì nó sẽ được đưa vào cùng một tệp sử dụng lập trình python để nhận diện hình ảnh qua các hình ảnh trong huấn luyện model cũng như dự đoán các hình ảnh ngoài để đánh giá hiệu suất chính xác mà tệp huấn luyện mang lại.

**Bước 4: Viết mã Python cho nhận diện biển báo và điều khiển LED**

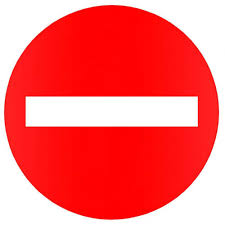
Viết mã Python sử dụng thư viện OpenCV để lấy nguồn hình ảnh chứa biển báo để nhận diện.

Sử dụng YOLOv8 để nhận diện các biển báo giao thông. Các biển báo mà mô hình YOLOv8 sẽ nhận diện trong dự án này là:



Hình 2: Cảnh báo có người đi bộ

Hình 1: Cấm đi ngược chiều



Hình 2: Cảnh báo giới hạn 10 tấn



Khi phát hiện biển báo cấm đi ngược chiều thì máy tính gửi tín hiệu điều khiển từ Python đến Arduino thông qua serial port để bật led đỏ tương tự xe sẽ không thể tiếp tục đi, khi nhận diện biển báo cảnh báo người đi bộ thì sẽ bật led vàng nghĩa là xe sẽ đi chậm, và khi gặp biển báo 10t thì bật led xanh nghĩa là xe sẽ tiếp tục di chuyển bình thường. Dưới đây là code python dùng để nhận diện và gửi tín hiệu:

**from ultralytics import YOLO**

**import serial**

**import time**

**sources = "0"**

**ser = serial.Serial('COM4',9600)**

**time.sleep(2)**

**model = YOLO("E:/AI\_learning/learning\_2/test\_train\_yolov8/runs/detect/train8/weights/best.pt")**

**results = model.predict(stream=True, imgsz=512, source=sources,show=True) # source already setup**

**names = model.names**

**for r in results:**

**for c in r.boxes.cls:**

**dis\_name = names[int(c)]**

**print(dis\_name)**

**if dis\_name == 'P.102':**

**ser.write(b'1')**

**elif dis\_name == 'W.224':**

**ser.write(b'2')**

**elif dis\_name == 'P.127':**

**ser.write(b'3')**

**else:**

**ser.write(b'0')**

Mã Arduino để nhận tín hiệu điều khiển từ Python và điều khiển các LED tương ứng với biển báo phát hiện được.

**int led\_red = 2;**

**int led\_yellow = 3;**

**int led\_green = 4;**

**void setup() {**

**Serial.begin(9600);**

**pinMode(led\_red, OUTPUT);**

**pinMode(led\_yellow, OUTPUT);**

**pinMode(led\_green, OUTPUT);**

**digitalWrite(led\_red, LOW);**

**digitalWrite(led\_yellow, LOW);**

**digitalWrite(led\_green, LOW);**

**// put your setup code here, to run once:**

**}**

**void loop() {**

**if (Serial.available() > 0){**

**char command = Serial.read();**

**if (command == '1'){**

**digitalWrite(led\_red, HIGH);**

**digitalWrite(led\_yellow, LOW);**

**digitalWrite(led\_green, LOW);**

**}**

**else if (command == '2'){**

**digitalWrite(led\_red, LOW);**

**digitalWrite(led\_yellow, HIGH);**

**digitalWrite(led\_green, LOW);**

**}**

**else if (command == '3'){**

**digitalWrite(led\_red, LOW);**

**digitalWrite(led\_yellow, LOW);**

**digitalWrite(led\_green, HIGH);**

**}**

**else if (command == '0'){**

**digitalWrite(led\_red, LOW);**

**digitalWrite(led\_yellow, LOW);**

**digitalWrite(led\_green, LOW);**

**}**

**}**

**// put your main code here, to run repeatedly:**

**}**

**III.Kết quả và Thảo luận**

**Kết Quả:**

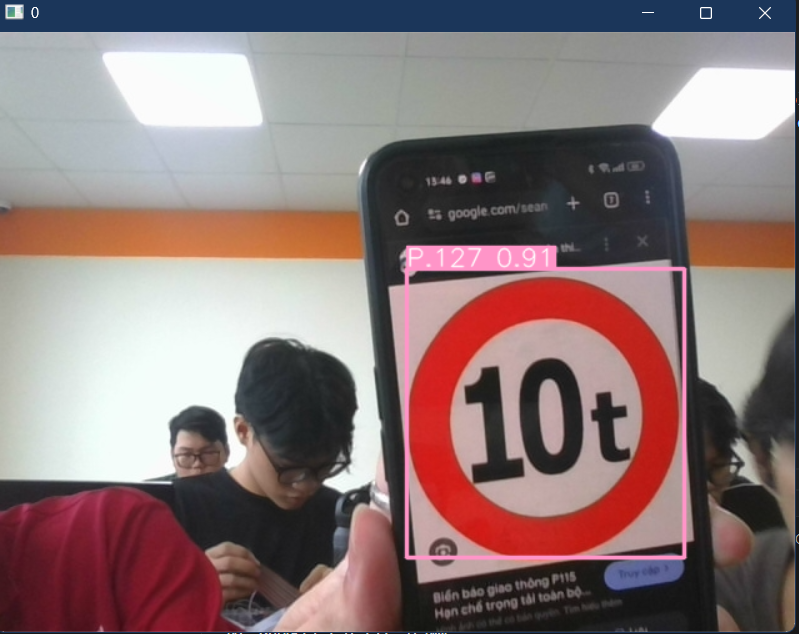
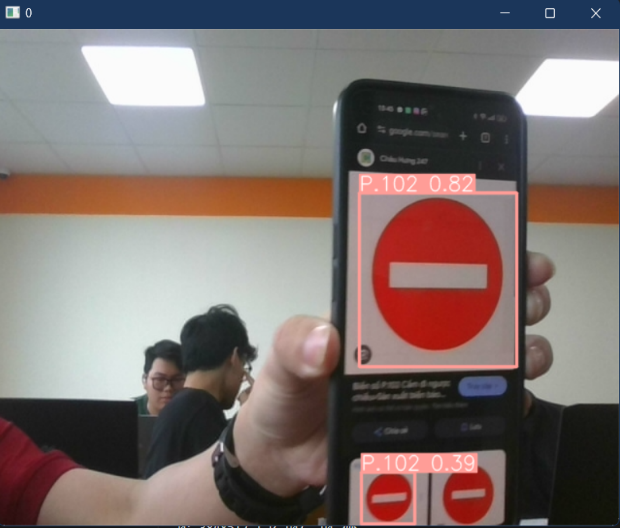
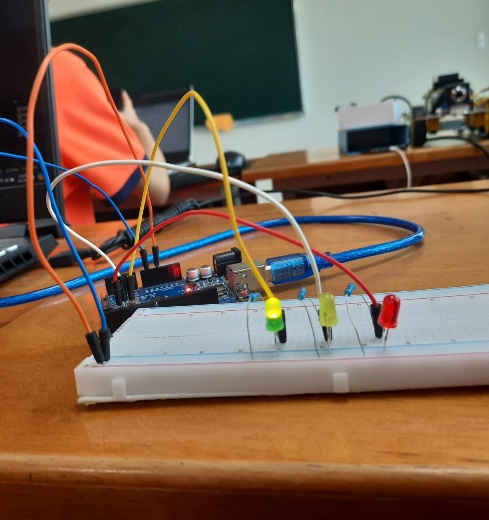
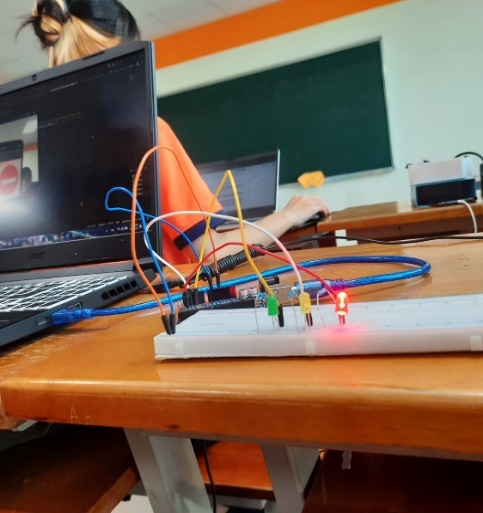
Dự án đã hoàn thành mục tiêu xây dưng được một hệ thống nhận diện biển báo giao thông đơn giản sử dụng những hình ảnh khác nhau để đưa ra tín hiệu phân biệt giữa chúng bằng thiết bị ngoại vi Arduino và led, sau đây là đánh giá những nội dung giúp cho kết quả của dự án hoàn thiện:

-Nhận Diện Biển Báo: Sử dụng YOLOv8, một mô hình deep learning hiệu quả trong việc nhận diện vật thể, hệ thống đã có khả năng nhận diện và phân loại các biển báo giao thông từ hình ảnh đầu vào với độ chính xác cao.

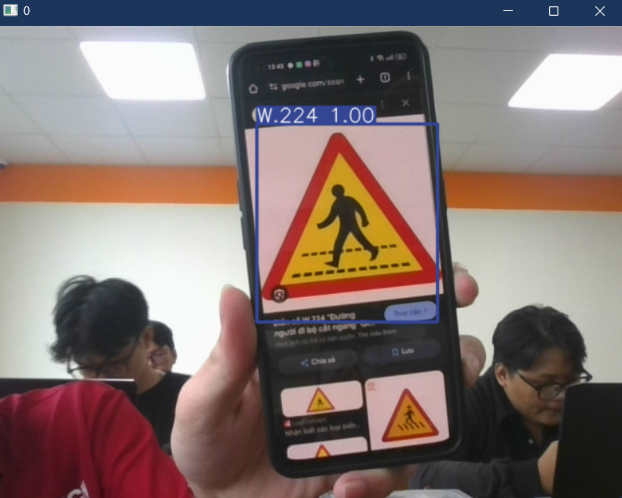
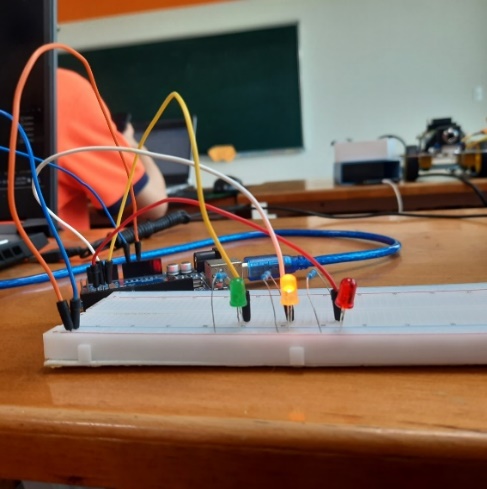
-Điều Khiển Led để phân biệt từng hình ảnh: Sử dụng Arduino, hệ thống có khả năng điều khiển các đèn LED tương ứng với biển báo giao thông phát hiện được. Khi nhận được tín hiệu từ phần mềm nhận diện biển báo, Arduino sẽ thực hiện thay đổi trạng thái của các đèn LED mô phỏng các trạng thái giao thông khác nhau.

-Giao Tiếp Giữa Python và Arduino: Sử dụng thư viện Arduino-Python API, pyserial, giúp việc giao tiếp giữa Python và Arduino thông qua cổng serial port đã được thực hiện một cách hiệu quả, cho phép truyền tín hiệu điều khiển một cách nhanh chóng và đáng tin cậy.

Dưới đây là những hình ảnh lấy ra từ kết quả của dự án:



**Thảo Luận:**



Mô hình YOLOV8 đã giúp cho hiệu quả nhận diện hình ảnh theo thời gian thực đạt độ chính xác cao nhất nhờ đó việc nhận diện được các hình ảnh biển báo giao thông chính xác từ cả trong tệp huấn luyện và tệp kiểm tra đã giúp dự án làm cho việc điều khiển led để phân biệt hình ảnh tốt hơn. Từ đó cho thấy hệ thống đã đạt được một mức độ hiệu suất và độ tin cậy đáng kể trong việc nhận diện biển báo và điều khiển đèn giao thông. Tuy nhiên, việc cải thiện hiệu suất và độ tin cậy vẫn là một điểm cần tiếp tục nghiên cứu và phát triển, đặc biệt là trong điều kiện ánh sáng yếu hoặc môi trường giao thông phức tạp.

Phần mềm python là ngôn ngữ lập trình giúp mở camera trên hệ điều hành Windows để nhận diện vật thể, ngoài ra với thư viện Pyserial đã làm cho việc giao tiếp giữa Arduino với máy tính trở nên hiệu quả hơn. Có thể nói hệ thống được xây dựng trên nền tảng phần cứng và phần mềm phổ biến, dễ dàng triển khai trong nhiều ứng dụng thực tế. Tuy nhiên, việc đảm bảo tính ổn định và bảo trì hệ thống trong thời gian dài vẫn là một thách thức đối với các dự án giao thông thông minh.

Với việc mở rộng các công nghệ khác để giúp cho việc nhận diện biển bảo giao thông đạt độ chính xác tuyệt đối để được ứng dụng trong cuộc sống thì hệ thống phải có khả năng tương thích và mở rộng với các công nghệ và phần mềm mới, không chỉ sử dụng ở led để cho người dùng thấy sự khác biệt giữa các biển báo mà có thể liên kết với những động cơ thực cũng như những công cụ khác phù hợp trong ứng dụng ngoài xã hội. Việc sử dụng ngôn ngữ và công cụ lập trình phổ biến như Python và Arduino cũng giúp dễ dàng tích hợp các tính năng và mở rộng chức năng trong tương lai.

Tóm lại, dự án đã đạt được kết quả tích cực trong việc xây dựng một hệ thống nhận diện biển báo và điều khiển đèn giao thông. Tuy nhiên, việc tiếp tục nghiên cứu và phát triển là cần thiết để cải thiện hiệu suất và tính ứng dụng của hệ thống trong các ứng dụng giao thông thực tế.

**Các tài liệu tham khảo:**

Cài đặt ROS**:** [Cài đặt ROS Noetic (robodev.blog)](https://robodev.blog/cai-dat-ros-noetic)

Giao thức truyền thông của máy tính điều khiển: [Những giao thức mạng bạn nên biết (bkaii.com.vn)](https://bkaii.com.vn/tin-tuc/621-nhung-giao-thuc-mang-ban-nen-biet)

Phân biệt MPU và MCU: [Vi điều khiển và vi xử lý | Tuan Anh IPC (wordpress.com)](https://tuananhipc.wordpress.com/2014/04/22/vi-dieu-khien-va-vi-xu-ly/)

Các ứng dụng Robot tích hợp AI trong thực tế:

Hình ảnh:

Xe tự lái: [Tesla đã sẵn sàng cho xe tự lái vào năm sau (thanhnien.vn)](https://thanhnien.vn/tesla-da-san-sang-cho-xe-tu-lai-vao-nam-sau-1851264765.htm)

Robot dịch vụ: [Covid-19 thúc đẩy tự động hóa ngành dịch vụ ăn uống - Nhịp sống kinh tế Việt Nam & Thế giới (vneconomy.vn)](https://vneconomy.vn/covid-19-thuc-day-tu-dong-hoa-nganh-dich-vu-an-uong.htm)

Robot y tế: [12 loại hình robot ứng dụng trong môi trường bệnh viện - Bệnh viện quận 11 (benhvienquan11.vn)](https://benhvienquan11.vn/so-y-te-tpho-chi-minh/12-loai-hinh-robot-ung-dung-trong-moi-truong-benh-vien-n1184.html)

Robot sản xuất: [Ứng dụng robot trong ngành công nghiệp ô tô (intechvietnam.com)](https://intechvietnam.com/tin-tuc/tin-cong-nghe/ung-dung-robot-trong-nganh-cong-nghiep-o-to-494/)

Robot giáo dục: [Robot sẽ thay thế vị trí của giáo viên? - Báo Cần Thơ Online (baocantho.com.vn)](https://baocantho.com.vn/robot-se-thay-the-vi-tri-cua-giao-vien-a90379.html)

Robot an ninh: [Nhật Bản ra mắt robot tuần tra an ninh tại sân bay - Tuổi Trẻ Online (tuoitre.vn)](https://tuoitre.vn/nhat-ban-ra-mat-robot-tuan-tra-an-ninh-tai-san-bay-20190531135823738.htm)

Nguồn dữ liệu dùng để huấn luyện: <https://www.kaggle.com/datasets/lvnduy/street-traffic-signs-in-vietnam-yolov8>

Nguồn tham khảo để huấn luyện dữ liệu: https://docs.ultralytics.com/usage/python/